

APPLICATION FOR UNITED STATES LETTERS PATENT

FOR

**METHOD AND SYSTEM FOR DIRECT SERVER  
SYNCHRONIZATION WITH A COMPUTING DEVICE**

Inventors:

Jeffrey Huang  
Magnus Vejlstrup  
Anupam Singhal  
Emerald Chung  
David Rabbers  
Martin Susser

Prepared by:

Blakely, Sokoloff, Taylor & Zafman  
1279 Oakmead Parkway  
Sunnyvale, California 94086  
(408) 720-8300

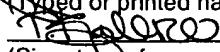
"Express Mail" mailing label number EL672752774US

Date of Deposit March 28, 2001

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Patricia A. Balero

(Typed or printed name of person mailing paper or fee)

  
(Signature of person mailing paper or fee)

SKD

005306.P007

# **METHOD AND SYSTEM FOR DIRECT SERVER SYNCHRONIZATION WITH A COMPUTING DEVICE**

## **FIELD OF THE INVENTION**

**[0001]** This invention relates to computer systems, and more specifically to a method and system for direct server synchronization with a computing device.

## **BACKGROUND OF THE INVENTION**

**[0002]** Palm or handheld computers have offered very limited functionality and have been restricted to being personal organizers. These watered-down computers are not compatible with programs on larger desktop and notebook computers, but rely on proprietary operating systems and programs to ensure small memory footprint and quick response time, even on low-power microprocessors. These restrictions have limited the usefulness of the personal organizers, as they operate stand alone and cannot readily share data with other computers.

**[0003]** Some handheld devices are capable of interacting with enterprise business applications running remotely on enterprise servers via a network, such as the internet. The two platforms were linked via one or more communication paths usually through a host server, including paths through a modem, a parallel port, a serial port, or a cradle assembly. Further, various wireless communication techniques, such as radio or infrared communication may be used. In the event that a cradle was used, when the handheld computer was in

the cradle and actively connected to the host computer, the handheld computer typically entered a mode to update data in the host computer and itself.

**[0004]** A major problem exists in the current methods of synchronizing data between the enterprise server and the palmtop or handheld computer. For instance, certain data such as calendar information, appointment information, and notes, among others, would generally be kept on the server, which may be networked so that other handhelds or workstations may remotely update the calendar information, appointment information, and notes. The same information was also kept on the palmtop or handheld computer for the user to review and update when mobile. However, if the user was traveling and information was entered on the server or on a remote station on a network server machine while the user was not at his or her office, or if the user entered information on the local version of the data in the palmtop or handheld computer, the information contained in the server and the palmtop or handheld computer would become non-coherent, leading to undesirable problems such as meeting conflicts and miscommunications.

**[0005]** Prior art methods synchronize a list of records, such as contact and appointments. They are not shared among different users and/or user accounts and only visible to the owner and can only be modified by the owner.

**[0006]** Some enterprise applications can handle very complex data for enterprises targeted for different areas, such as Field Service and Sales Force Automation, and more. The data models in applications are much more rich than

those which can be represented by list of records. Plus, the data may have the following visibilities:

- 1) Visible only to particular person,
- 2) Visible only to a particular position (each person may have one or more positions in an organization),
- 3) Visible to a group of persons, such the persons in the same sales team, or in the same account team,
- 4) Visible to management positions,
- 5) Visible to a particular organization in a company,
- 6) Visible across all organizations in a company.

**[0007]** The change in the visibility of a particular piece of data leads to the changes of the scope of people who can see and change it. If more than one person can change it, the prior art methods of synchronization failed to keep the data coherency among all the persons who can see and modify it.

**[0008]** The prior resolutions of the data synchronization problem were generally very time consuming and tedious and therefore reduced the effectiveness of the palmtop/desktop combination. One solution was to limit the accessibility of files such that only files physically residing with the user were the master. However, this solution dictated that no changes could be made to the other system without the danger of losing those changes. Thus, while the palmtop/desktop combination of computers existed, the use of the combination

was not popular due to the requirement that the user had to manually synchronize the data on both platforms. Therefore, the problem of synchronization limited the potential of the palmtop or handheld computer market.

**[0009]** One problem encountered by the current generation of handheld devices and their add-on connectivity solutions is that the connectivity solutions rely on batch mode rather than real-time oriented approaches to data coherency. Because the server and the handheld device typically became disconnected and incommunicado when the user goes mobile, the data synchronization problem becomes intractable. During the mobile period, the creation, the deletion and the modification of data causes incoherency that often results in conflicts reconcilable only through manual intervention by the user. These conflicts would arise because the server and handheld device were unable to communicate changes in their data until they were reconnected using a serial, parallel or modem connection.

**SUMMARY OF THE INVENTION**

[0010] In one embodiment, the method comprises synchronizing directly a computing device and an enterprise server, which comprises retrieving a record extraction sequence from the server; and extracting records stored on a database according to the record extraction sequence, wherein the extracted records are not already stored on the computing device.

**BRIEF DESCRIPTION OF THE DRAWINGS**

[0011] The accompanying drawings, which are included as part of the present specification, illustrate the presently preferred embodiment of the present invention and together with the general description given above and the detailed description of the preferred embodiment given below serve to explain and teach the principles of the present invention.

[0012] **Figure 1** illustrates a block diagram of a direct server synchronization system 100.

[0013] **Figures 2A and 2B** illustrate an exemplary flow diagram of the data exchange between handheld 110 and server 130 via web engine 120 for direct server 130 synchronization.

[0014] **Figure 3** illustrates a computer system 300 representing an integrated multi-processor, in which elements of the present invention may be implemented.

## DETAILED DESCRIPTION

**[0015]** A method and system for direct server synchronization with a handheld device is disclosed. In one embodiment, the method comprises synchronizing directly a handheld device and an enterprise server, which comprises retrieving a record extraction sequence from the server; and extracting records stored on a database according to the record extraction sequence, wherein the extracted records are not already stored on the handheld device.

**[0016]** In the following description, for purposes of explanation, specific nomenclature is set forth to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention.

**[0017]** Some portions of the detailed descriptions which follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven

convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

**[0018]** It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

**[0019]** The present invention also relates to apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access

memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

**[0020]** The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein.

**[0021]** **Figure 1** illustrates a block diagram of a direct server synchronization system 100. System 100 includes handheld device 110. The present method and system is not limited to handheld device 110. It may be applied for the direct server synchronization of computing machines, like personal computers and laptops as well. Handheld 110 could be a Palm™ operating system device such as those manufactured by Palm, Inc. of Santa Clara, California, or a Microsoft CE™ operating system device, such as those manufactured by Casio, Inc. of Dover, New Jersey, or similar Palm and CE devices. Included in handheld 110 is synchronization client 111 (Sync Client). Sync client 111

- a) provides a user with a synchronization interface ;
- b) interfaces with modules, such as a handheld database generator, a handheld transaction manager, and a data integrity manager;
- c) completely rollsback transactions after the previous synchronization;
- d) imports the net changes downloaded to handheld 110 so that the local data model on sync client 111 reaches the same state as that of database 140;
- e) sends log off requests to server 130;
- f) sends a request to server 130 for a device ID if it is the first time handheld 110 is synchronized with a server 130;
- g) sends its device ID to server 130 if it has previously synchronized with server 130; and
- h) sends a request to server 130 to download application definitions (meta-data) if it is the first synchronization, there are meta-data updates caused by responsibility changes, server 130 upgrades the application with new meta-data, or meta-data is missing on sync client 111.

**[0022]** Each device 110 synchronizing with a server 130 hosting an enterprise application has a unique ID. Server 130 uses the device ID to determine performance requirements, security and licensing management.

SKD 950 950 950 950 950 950 950 950

[0023] Web Engine (WE) 120 is coupled to handheld 110. WE 121 includes a WE proxy plug-in 121. Plug-in 121 is a Hyper Text Transport Protocol (HTTP) extension which translates HTTP requests to a different networking protocol, such as SISNAPI. SISNAPI is a protocol operating on top of TCP/IP, which is similar to HTTP, but provides additional performance in data transport, security, and scalability.

[0024] Server 130 is connected to WE 120 and may be an enterprise server running enterprise applications or similar server. Server 130 includes a synchronization engine 131 (Sync Engine) which communicates with plug-in 121. Sync Engine 131

- a) handles requests for logging-in and logging-out from sync client 111;
- b) uploads transactions from handheld 110 and processes those transactions;
- c) extracts records for business components from database 140;
- d) packs and sends records to handheld 110;
- e) generates a digital signature for checking synchronization integrity; and
- f) Finds the net changes of the sync client 111 based on its synchronization history.

[0025] Sync Engine 131 includes a server sync manager (SSM). SSM is comprised of many sub-managers, including a transaction processing manager,

a report manager, a data extraction manager, a transaction history manager (THM), a user profile manager (UPM), a conflict resolution manager (CRM), a meta-data sync manager (MSM), and a sync visibility manager (SVM).

**[0026]** The transaction processing manager parses the transactions sent from SSM and executes the transactions one-by-one. The report manager reports conflicts, transaction errors, and synchronization results. The data extraction manager accepts requests for extracting raw data, and packs the data into a text file in the form of an object. The object is defined by a universal object locator which uniquely identifies each object.

**[0027]** The data extraction manager manages the integrity and consistency of synchronized raw data. THM stores a transaction history and compares the results for each sync session. THM stores the results and works with the report manager and data extraction manager.

**[0028]** UPM manages user profiles, such as usernames, application names, device, object manager names, and sync-filters used by the user. CRM cooperates with report manager and the transaction processing manager to ask for feedback and confirmation when conflicts occur.

**[0029]** MSM extracts meta-data (such as screen, view, and field definitions) and packs the meta-data into a XML format. The meta-data is stored locally in server 130. Additionally, MSM monitors the consistency and integrity of meta-data between server 130 and handheld 110. MSM supports repository upgrades, and monitor's responsibility changes that lead to meta-data updates.

DRAFT - 6/12/2009 10:22:50 AM

[0030] The SVM applies rules for cutting down the amount of data synchronized between server 130 and handheld 110.

[0031] Server 130 also includes object manager 132 which loads objects in and out of database 140. Also included in system 100 is database 140. Database 140 may be a relational database, such as an Oracle<sup>TM</sup> database, or similar storage device.

[0032] System 100 allows handheld 110 to synchronize directly with server 130, over the internet, serial connection dial-up, Ethernet, infrared wireless, or other similar network, including enterprise data having multiple relationships. The present system and method resolves conflicts between server 130 and handheld 110 that occur during synchronization. The present techniques ensure synchronized data is reliable on both handheld 110 and server 130.

[0033] **Figures 2A and 2B** illustrate an exemplary flow diagram of the data exchange between handheld 110 and server 130 via web engine 120 for direct server 130 synchronization. The process begins at start block 200. At processing block 201 handheld 110 logs-in to server 130. Logging-in may be initiated by the user selecting a synchronization button on handheld 110. At decision block 202, server 130 confirms that handheld is allowed access to server 130. If the login is not valid, then flow ends at stop block 299. If the login is valid, flow continues to processing block 203.

PROCESSED - DRAFT

**[0034]** Handheld 110 retrieves a synchronization identifier (Sync ID) from server 130, at block 203. The sync ID is a unique identifier assigned to handheld 110. No two devices across the enterprise will have the same sync ID. The sync ID is never reused, reassigned, or recycled. Finally, the sync ID is stored on handheld 110. Flow continues to processing block 204.

**[0035]** Handheld 110 requests and retrieves a repository version number or code from server 130. The repository version allows handheld 110 to determine if it needs to synchronize by comparing the repository version with a locally stored repository version retrieved from its previous sync session. At processing block 205, handheld 110 retrieves views from server 130. A view is a special display of data, created as needed. A view temporarily ties two or more files together so that the combined files can be displayed, printed or queried. All accessible views and default views for the user of handheld 110 are retrieved.

**[0036]** Default views specify views in an application to be synchronized and transferred to handheld 110. Applications running on handheld 110 may share common application definitions with those running on other platforms, such as desktops and laptops. Default views provide a way to limit the volume of data transferred to handheld 110.

**[0037]** Furthermore, default views specify all the views that server 130 will extract from database 140 and download and synchronize with handheld 110. Each view has one business object associated with it. Each view may have a number of applets and each applet has a business component associated with it.

[0038] Accessible views specify the views that a particular user account can see. It is controlled by responsibilities (a processing step performed by an object). Each user account can have one or more responsibilities. And each responsibility has a number of assigned views, which can be configured by an application. The accessible view list further limits the views a handheld 110.

[0039] Default Business Object lists specify all the extra data used by handheld 110 at the application level and/or special customization of business logics at the business component level.

[0040] PDQ views result from PDQs that serve as synchronization filters to limit the volume of data transferred to handheld 110. PDQs are defined by a system administrator.

[0041] Flow continues to processing block 206 where handheld 110 retrieves a list of business objects from server 130. If any objects in the list are missing from handheld 110, they are retrieved from server 130.

[0042] At processing block 207, handheld 110 retrieves a list of previous transactions from server 130. The previous transactions list lists the transactions previously synchronized on handheld 110. Handheld 110 retrieves the list of previous transactions from server 130 to verify if server 130 in fact received it. Handheld 110 stores the list locally in case the local list is different from that retrieved from server 130. It implies that server 130 does not completely receive and handheld 110 should send the list of previous transactions again to server 130. Afterwards, handheld 110 will retrieve the transactions from a local

DRAFT DRAFT DRAFT

database, pack the transactions, and use them to replace the local list. This will be used in the next synchronization session and sent to server 130.

**[0043]** With the list, handheld 110 may determine which transactions need to be processed. Flow continues to processing block 208, where server 130 processes any necessary transactions, determined in processing block 207. Each transaction is processed individually and a confirmation is sent to handheld 110 for each transaction, until all transactions are processed.

**[0044]** At processing block 209, handheld 110 retrieves an event list from server 130. The event list allows handheld 110 to synchronize all events that occurred since the last synchronization with server 130. Flow continues to processing block 210 where handheld 110 retrieves the PDA repository from server 130. A repository upgrade and responsibility change may be detected. If the repository is upgraded, a new PDA master repository is downloaded from handheld 110 to server 130. On server 130, the PDA repository will be regenerated based on the new PDA repository.

**[0045]** At processing block 211, handheld 110 sets PDQs on server 130. Thus, once server 130 sends handheld 110 a confirmation, the PDQs are synchronized on both sides. Flow continues to processing block 212 where handheld 110 retrieves an extraction sequence ID from server 130. The extraction sequence ID allows handheld 110 to know which database records need to be extracted from database 140 in order to be synchronized with server 130.

P005306.P007

**[0046]** Flow continues to processing block 213 where handheld 110 extracts necessary records from database 140 and returns a confirmation to server 130. At decision block 214, handheld 110 determines if the Extraction Sequence ID obtained at processing block 212 matches the previously obtained Extraction Sequence ID. If the Sequence IDs match, then flow continues to processing block 215. If the Sequence IDs do not match, then flow continues to processing block 216. At processing block 215 handheld 110 performs a "net" data extraction from Server 130. The net changes contain only the changes in database 140 since the last synchronization, which are relevant to handheld 110. A full data extraction from server 130 contains a snapshot of database 140, which is relevant to handheld 110. A full data extraction is retrieved at processing block 216.

**[0047]** Flow continues to processing block 217, from both processing blocks 215 and 216, where handheld 110 completely rolls back transactions that occurred subsequent to the last synchronization.

**[0048]** At processing block 218, handheld 110 imports the net changes downloaded to the device so that the local data model on sync client 111 reaches the same state as that of database 140.

**[0049]** Flow continues to processing block 219 where handheld 110 logs off server 130 and the direct server sync process completes at stop block 299.

**[0050]** All request are sent between handheld 110 and server 130 using HTTP. Parameters passed from handheld 110 so server 130 may use HTTP

post parameters, such as username and pass word for logging-in, paced transactions, application names and sync filters.

**[0051]** **Figure 3** illustrates a computer system 300 representing an integrated multi-processor, in which elements of the present invention may be implemented. For example, system 300 may be the architecture of server 130 or handheld 110. One embodiment of computer system 300 comprises a system bus 320 for communicating information, and a processor 310 coupled to bus 320 for processing information. Computer system 300 further comprises a random access memory (RAM) or other dynamic storage device 325 (referred to herein as main memory), coupled to bus 320 for storing information and instructions to be executed by processor 310. Main memory 325 also may be used for storing temporary variables or other intermediate information during execution of instructions by processor 310. Computer system 300 also may include a read only memory (ROM) and/or other static storage device 326 coupled to bus 320 for storing static information and instructions used by processor 310.

**[0052]** A data storage device 327 such as a magnetic disk or optical disc and its corresponding drive may also be coupled to computer system 300 for storing information and instructions. Computer system 300 can also be coupled to a second I/O bus 350 via an I/O interface 330. A plurality of I/O devices may be coupled to I/O bus 350, including a display device 343, an input device (e.g., an alphanumeric input device 342 and/or a cursor control device 341).

[0053] The communication device 340 is for accessing other computers (servers or clients) via a network. The communication device 340 may comprise a modem, a network interface card, or other well known interface device, such as those used for coupling to Ethernet, token ring, or other types of networks.

[0054] A method and system for direct server synchronization with a handheld device is disclosed. Although the present invention has been described with respect to specific examples and subsystems, it will be apparent to those of ordinary skill in the art that the invention is not limited to these specific examples or subsystems but extends to other embodiments as well. The present invention includes all of these other embodiments as specified in the claims that follow.